

Bit-map-based Spatial Data Transmission Scheme

Gi Oug OH*

Abstract

This paper proposed bitmap based spatial data transmission scheme in need of rapid transmission through network in mobile environment that use and creation of data are frequently happen. Former researches that used clustering algorithms, focused on providing service using spatial data can cause delay since it doesn't consider the transmission speed. This paper guaranteed rapid service for user by convert spatial data to bit, leads to more transmission of bit of MTU, the maximum transmission unit. In the experiment, we compared arithmetically default data composed of 16 byte and spatial data converted to bitmap and for simulation, we created virtual data and compared its network transmission speed and conversion time. Virtual data created as standard normal distribution and skewed distribution to compare difference of reading time. The experiment showed that converted bitmap and network transmission are 2.5 and 8 times faster for each.

▶ Keyword: Location-Based Service, Spatial Data, Grid, Bit-Map, R-Tree

I. Introduction

ICT와 GPS 기술이 발전하면서 대량 생성되는 위치 정보의 생성과 규모는 빅 데이터를 구성하고 있다. 개인의 위치 정보는 특정 지역에서 촬영한 사진의 태그, 이동통신 단말기를 사용한 카드 결제에 의한 위치 정보, 통화를 위한 발신 정보, SNS 및 이동 위치를 따라 GPS에서 다량으로 수집되며 시간별 위치 기반의 이동 궤적을 구성한다. 이와 같은 위치 정보는 개인적으로 이동 이력 관리나 생활방식의 공유 등에 사용될 수 있으며 다른 사람에게는 위치 공유를 통해 특정 지역을 추천하거나 정보를 제공할 수 있다.

위치기반서비스(LBS, Location-Based Service)는 GPS의 위치 정보를 기반으로 사용자들에게 관심 사항이나 주요 서비스를 제공하면서 상용 적으로 활용되고 있다. 특히, 최근에는 상품의 정보뿐만 아니라 각 지역의 날씨, 온도, 미세먼지의 지역적 분포, 조류 인플루엔자와 같은 전염병의 발생 위치, 화재나 응급 재난 발생 시 해당 위치에 대한 정보를 전달하는 주요 수단으로 활용된다[1][2].

시간과 공간으로 표현되는 위치 정보는 지도의 공간적 위치 [3]뿐만 아니라 이미지의 색 공간[4], 입체 도형이나 공중의 위치 등을 표현할 수 있다.

본 논문에서는 지도를 이용한 지리적 위치 정보를 대상으로 한다. 지리적 위치 정보는 사용자의 이동이나 각종 질병 및 환경정보를 대상으로 이용되기도 하며 이러한 특성을 반영하기 위해서는 공간적 클러스터링 알고리즘을 사용한다[5]. 공간 클러스터링은 공간 객체에 대한 그룹화로 인공지능의 기계학습에서 사용되며 일차적으로 데이터 포인트(Data point)가 속한 지역의 그룹을 확인함으로써 데이터 포인트와 관련된 객체를 찾고 사용자에게 전송하여 서비스를 제공하는 기본적인 알고리즘이다. 클러스터링으로 분석된 공간 객체는 서버에 저장되고 사용자에게 네트워크를 통해 전송된다. 클러스터 객체를 찾는 방법으로는 분할 기반 클러스터링[6], 그래프 기반 클러스터링 [7], 그리고 그리드 기반 클러스터링 기법[8]이 있다. 이들은 두 개의 관점을 강조하는 알고리즘인데, 하나는 지리적 공간 데이터가 빅 데이터에 해당하므로 포인트를 기준으로 신속한 클러스터 구성을 주장하고 있다. 다른 하나는, 알고리즘의 실용적인 적용을 강조한다.

공간 객체의 클러스터링은 사용자에게 제공되는 서비스라는 관점에서 현실적인 요구를 반영한 주요 기법이다. 따라서, 공간

• First Author: Gi Oug OH, Corresponding Author: Gi Oug OH

*Gi Oug OH (king5@gachon.ac.kr), Dept. of Computer Engineering, Gachon University

• Received: 2019. 07. 17, Revised: 2019. 08. 23, Accepted: 2019. 08. 25.

에 분포된 연관 객체들을 신속하게 분석하는 것은 서비스의 신속성을 보장할 수 있어서 연구할 가치가 있다. 공간 객체는 지리적 위치, 거리뿐만 아니라 각 객체의 속성까지도 분석 대상이며 클러스터링의 주요 파라미터(Parameter)이다. 다양한 파라미터를 기준으로 분석된 객체들은 데이터 포인트와 함께 사용자에게 전송된다. 이 과정에서 클러스터링 결과가 신속하게 전달되기 위해서는 빅 데이터를 분석하는 클러스터링 알고리즘의 처리 속도가 빨라야 하지만, 클러스터링 그룹에 대한 위치 정보를 신속하게 전송하는 것도 중요하다. 현재까지, 위치기반서비스나 공간 객체 클러스터링에서는 공간 객체를 식별하고 그룹화하는 알고리즘에 대해서만 신속한 처리를 고려하였다. 그러나 서버에서 알고리즘의 효율성이 강조되더라도 네트워크로 전송되는 데이터의 규모가 크다면 사용자의 서비스 응답속도는 늦어진다.

위치기반서비스나 인공지능의 클러스터링에서 공간 객체의 개별 위치나 객체들의 영역 정보는 서비스 측면에서 제공되기 때문에 사용자에게 신속하게 제공되어야 한다. 그러나 기존의 공간 객체 식별 기법[9]이나 클러스터링 기법[10]은 자체적으로 신속한 공간 객체 분석 알고리즘을 제공하는 것이지만 이를 네트워크를 통해 신속히 전송하는 것까지 연구가 확장되지 않았다.

공간 객체는 R-tree에서 MBR로 추정하며 하나의 객체를 나타내기 위해서는 대각 모서리의 두 점 $(x_1, y_1)-(x_2, y_2)$ 을 사용한다. 두 점은 x축과 y축으로 표현되며 각 축은 정수형 또는 실수형으로 4바이트나 8바이트의 크기로 관리된다. 이는 네트워크에서 패킷의 MTU (Maximum Transmit Unit)를 많이 차지하기 때문에 클러스터링 영역 정보를 전송하는데 속도 측면에서 부담이 된다.

본 논문에서는 공간 객체 또는 클러스터링한 그룹을 네트워크로 신속하게 전송하기 위한 비트맵 기반의 공간 객체 전송기법을 제안한다. 공간 객체는 하나의 공간에 존재하므로 영역을 그리드로 양자화하여 공간 객체를 포함하는 그리드 셀을 객체의 정보로 전송하며 클러스터링에서는 객체들이 속해있는 그룹의 영역 정보를 전송한다. 제안기법은 네트워크에서 전송속도를 빠르게 하려고 그리드 기반의 공간 영역 정보를 비트맵으로 변환한다. 실험에서는 가상의 데이터 집합을 사용하여 개별 공간 객체와 클러스터링 된 객체 그룹의 정보를 네트워크로 전송하며 MTU 크기, 비트맵 변환 속도 및 이더넷(Ethernet)에서 전송속도를 비교하였다. 실험 결과는 크기와 전송속도에서 각각 작은 크기와 빠른 전송시간을 나타냈다. 이는 공간 객체 분석에 집중한 기존 알고리즘의 처리시간에 추가로 네트워크 전송시간이 빨라짐으로써 사용자에게 대한 신속한 서비스가 가능한 것을 보여준 것이다.

논문의 구성은 다음과 같다. 2절에서는 기존의 공간 객체 처리 기법들을 살펴보고 정보 전송기법의 유무를 확인한다. 3절에서는 공간 객체의 위치 정보를 빠르게 전송하기 위한 비트맵 기반의 공간 정보 표현 기법을 설명한다. 제안기법에 대한 성능

은 가상의 공간 데이터를 사용하여 시뮬레이션하며 성능에 대한 논의는 4장에서 진행한다. 본 논문의 결론은 5장에서 기술하며 논문을 마무리한다.

II. Preliminaries

1. Spatial Data Index : R-tree

지리적 공간 객체는 위도와 경도를 사용하여 2차원으로 표현된다. 그러나 실제 공간 객체는 다양한 다각형의 형태로 지표면에 존재하기 때문에 위도와 경도만으로는 정확한 위치를 표현하기에 부족하다. 또한, 각 공간 객체의 위치는 지리적으로 차지하는 면적에 따라 결정되어야 한다.

R-tree는 지리적 공간 객체를 색인으로 관리하기 위해 [11]에서 제안된 공간 색인이다. 공간 색인은 (Fig. 1)의 r5, r8, r11과 같이 다양한 모습의 공간 객체를 감싸는 최소경계 사각형(MBR, Minimum Bounding Rectangle)으로 표현하며 MBR 사이의 포함관계를 트리 구조로 나타낸다. 트리는 R0를 루트 노드로 하여 레벨 1은 R1, R2, R3이며 리프 노드(Leaf node)에는 공간 객체인 r4, ..., r12가 저장된다. 트리에서 중간 노드는 한 MBR에 대한 대각선의 두 점을 저장하여 면적을 나타낸다.

본 논문에서는 지리적 공간을 그리드로 나누고 그리드 셀과 MBR의 포함관계에 따라 셀에서 객체의 위치 및 영역을 계산한다. 이는 MBR과 그리드 셀이 사각형 구조이므로 계산을 위한 효율성이 가장 적합한 구조이다.

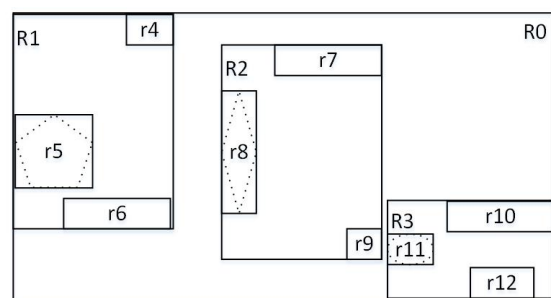


Fig. 1. MBRs of R-tree

2. Grid Based Spatial Division and Clustering

공간 데이터에 대한 빅 데이터 처리는 하둡(Hadoop)을 이용한 분산처리를 사용한다[9]. 일정한 영역에 대량의 공간 데이터가 분포되어 있으므로 지역별 공간 데이터의 분포를 식별하기 위해서는 영역을 구분해야 한다. 영역 구분 기법 중 그리드는 단순하면서 쉬운 기법에 해당한다.

[9]에서는 하둡(Hadoop) 기반의 클러스터링 된 서버에서 공간 데이터를 처리하기 위해 공간을 그리드로 구분하고 (Fig. 2)와 같이 각 서버에게 일정한 수의 그리드 셀 a, b, c, d를 할당하여

처리한 후 중앙 서버(Main node)에서 취합한다. 이러한 기법은 독립된 그리드 셀의 처리를 통해 각 서버의 부하를 줄이면서 병행 처리를 통해 전체 시간을 줄이는 효과가 있다. 최종적으로 수집된 공간 데이터는 중앙 서버에서 사용자들에게 전송되어 서비스로 제공된다. 앞의 논문에서는 하둡에 의한 공간 데이터의 분산처리에 초점을 맞추고 있으나 서비스로 제공될 공간 데이터의 전송 방법에 대해서는 논의되지 않았다. 결과적으로 서버에서 공간 데이터 처리는 효율적일 수 있으나 사용자에게 제공하는 서비스로는 데이터의 양에 따라 지연시간이 발생할 수 있어서 시간적 일관성이 있는 서비스 제공을 보장하기는 어려운 단점이 있다.

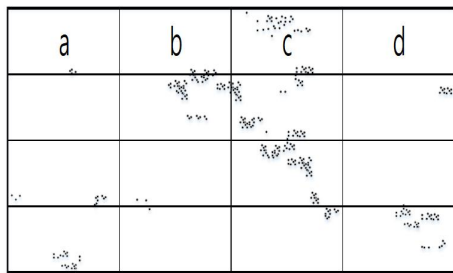


Fig. 2. Grid

그리드 기반 클러스터링[10]은 인공지능의 기계학습에서 공간 데이터를 위치나 속성으로 구분하여 묶는 기법이다. 공간 데이터를 클러스터링하면 한 공간에서 데이터 키를 중심으로 위치 또는 속성에 따라 같은 유형의 데이터별로 그룹화된다.

클러스터링 결과는 같은 종류의 데이터를 묶는 것이므로 사용자들에게 서비스로 제공됨으로써 의사결정을 위한 유용한 정보를 제공하게 된다. 논문 [10]은 의사결정자의 의사결정 과정을 돕기 위해 사용자들의 선호 매장 위치, 조류 인플루엔자의 분포 등에 대한 그리드 기반의 클러스터링을 사용한다(Fig. 3). 그리드의 장점은 영역을 단순하게 그리드로 구분하고 각 셀을 독립적으로 판단할 수 있다는 것이다. 이 논문은 클러스터링을 통해 유용한 연구결과를 제시하고 있으나 의사 결정자에게 분석 결과를 신속히 제공하는 방안은 간과하고 있다. 즉, 공간 데이터에 대한 유용한 분석 결과를 더욱 빠르게 제공하기 위한 연구가 추가된다면 의사결정에 시기 적절성(Time to Market)을 제공할 수 있다.

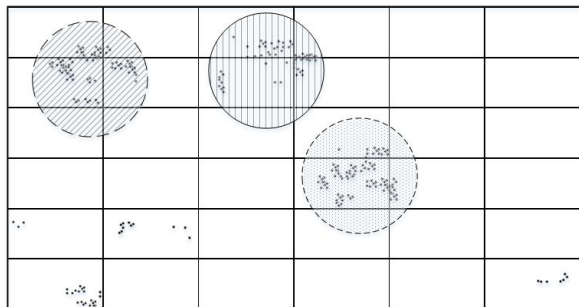


Fig. 3. Spatial Data Clustering Using Grid

III. Bit-Map of Spatial Region

공간 데이터를 네트워크로 전송할 때, TCP/IP의 고정된 MTU를 사용하는 경우에 신속한 전송을 위해서는 데이터 크기를 최소화해야 한다.

다음의 (Fig. 4)는 제안기법의 기본 개념을 도식화한 것이다. 먼저, 탐색 범위를 그리드로 나누고 각 그리드 선의 좌표 (x, y)를 위치 정보로 사용하면 실제 값보다 작은 값으로 같은 객체 및 그룹을 표현할 수 있다는 것이 본 논문의 기본 아이디어이다. 그룹 영역을 포함하는 그리드 셀은 두 대각 점으로 선택할 수 있으며 이를 비트맵으로 변환하면 객체가 최소의 크기로 표현된다.

지리적 공간 객체의 위도, 경도 표현은 서울의 은평구 갈현 1동의 경우 (37.6216493, 126.913107)이며 실수형에 저장된다. 지도의 임의의 점에서 일정한 거리에 있는 공간 객체를 Meter나 Kilo-meter로 나타내는 경우 그 값은 정수형에 저장된다.

(Fig. 4)에서 공간 객체 g는 x축의 그리드선 1, 2번 사이에 위치하며 y축으로는 5, 6번에 위치하므로 두 대각 좌표는 셀로 표현하여 (1, 5)-(2, 6)이다. 따라서 g의 원래 위도, 경도의 좌표보다 작은 값으로 공간 데이터의 위치를 나타낼 수 있다. 물론 g는 MBR로 표현되므로 그리드 셀을 사용하면 객체의 면적이 해당 셀의 선까지 확장되기 때문에 실제 면적과 차이가 발생하는 문제점이 있다. 이러한 문제점은 점 e와 같이 그리드 셀을 서브 그리드로 세분화하여 셀의 면적을 실제 공간 객체의 MBR과 일치시킬 수 있다.

클러스터링한 공간 데이터 f는 8개의 셀을 차지하고 있으므로 면적은 x축의 2, 4번 셀 선과 y축의 3, 7번 셀 선으로 나타내므로 (2, 3)-(4, 7)이다.

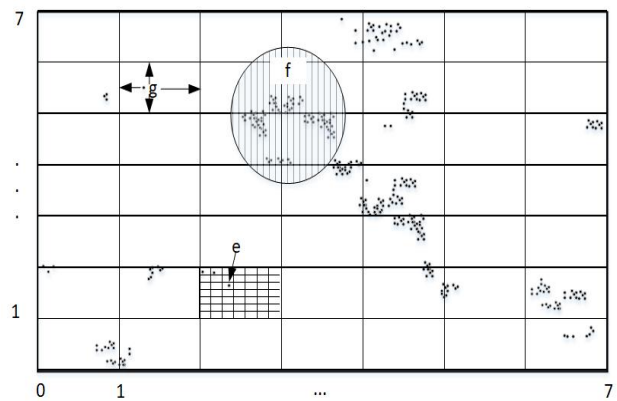


Fig. 4. Spatial Objects and Clustered Objects in a Grid

본 논문에서는 4바이트 이상을 차지하는 위치 정보를 그리드 선으로 대체함으로써 다음의 수식 (1)에 의해 비트로 표현한다. 수식 (1)에서 R은 지리적 공간을 나타내며 grid_numbers는 R의 각 축에 대한 그리드 개수이다. grid_units은 각 축에서 계산된 그리드 셀의 x, y축 크기이다.

$grid_units = R / grid_numbers$ ($R \geq grid_units$) (1)

지리적 공간을 그리드로 구분할 때의 특징을 휴리스틱 (Heuristic)으로 정리하면 다음과 같다.

휴리스틱 1. 지리적 공간 분할 시 복잡도 감소.

지리적 공간에서 공간 데이터의 분포를 조사하기 위해서는 각 공간 객체의 위치를 기준으로 상대 거리를 측정하여 근거리의 대상을 그룹화 하여야 한다. 이 과정에서 모든 공간 데이터를 반복적으로 비교하여 영역을 구분해야 하므로 시간 복잡도가 높다. 그러나 그리드로 공간을 구분하면 각 셀 단위로 공간 데이터를 관리하므로 비교 횟수가 감소한다.

휴리스틱 2. 클러스터링 된 객체 그룹의 쉬운 영역 설정.

공간 데이터의 거리나 특성을 기준으로 클러스터링 하면 각 객체의 구체적인 위치보다는 객체들이 포함된 영역이 사용된다. 그리드 셀은 공간 객체들의 범위가 되므로 셀을 사용하여 클러스터링의 영역을 쉽게 판단할 수 있다.

휴리스틱 3. 그리드 셀의 개수가 비트의 크기를 결정함.

수식 (1)과 같이 $grid_numbers$ 가 8이면 x, y축의 그리드는 8개가 생성된다. 8개의 그리드에서 각 그리드 선의 번호를 비트로 변환하기 위해서는 최소 4bit가 필요하다. 같은 이유로 256개의 그리드 선을 표현하기 위해서는 8bit가 필요하므로 그리드의 수에 따라 비트의 수는 증가한다.

(Fig. 4)는 그리드로 구분된 영역을 8개의 그리드 선으로 구분하였다. 공간 데이터의 위도와 경도는 (Fig. 4)와 같이 x, y축의 그리드 선이 만나는 곳을 기준으로 표현된다. 이는 (Fig. 5)의 마지막 행과 같이 각각 4bit로 변환되며 위도와 경도에서 8바이트인 좌표 값은 총 2바이트가 된다. 그리드의 분할을 더 세밀하게 하여 그리드 선이 많아지면 비트는 증가하지만, 여전히 비트로 표현되어 작은 값으로 표현할 수 있다.

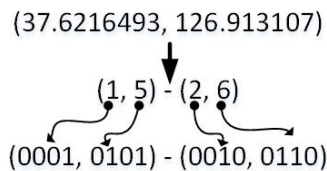


Fig. 5. Bit-map

(Fig. 6)은 비트맵 변환 알고리즘으로 개별 공간 객체에 대한 비트변환을 나타내고 있으며 Python 형식의 의사 코드 (pseudo code)이다. 클러스터링 된 영역의 비트맵 변환은 개별 객체 변환과 같은 알고리즘을 사용하므로 쉽게 변환할 수 있다.

알고리즘 Spatial BitMap은 인수로 지리적 공간의 Dataset과 그리드 분할을 위한 $grid_numbers$ 를 인수로 받는다. 입력 받은 공간 데이터 집합은 x, y축의 최솟값과 최댓값을 분석하여 공간의 크기를 결정한다(코드 01-02). 최솟값인 x_min ,

y_min 은 공간의 시작 좌표가 되며 공간에 대한 그리드 분할은 x_max , y_max 를 $grid_numbers$ 로 나누어 $n \times n$ 의 정방형으로 생성한다(코드 03-04).

한 공간 객체의 MBR은 그리드 셀에서 소스 코드 05번과 같이 왼쪽 x 좌표는 왼쪽 그리드 선(gridline)으로 확장되며 오른쪽 x 좌표는 오른쪽 그리드 선으로 확장된다. 같은 방식으로 y 좌표는 위와 아래로 확장된다. 이때, 그리드 선은 정수이므로 객체의 좌표를 $grid_units$ 로 나눈 결과는 바닥함수(floor)나 천정함수(Ceil)을 사용하여 변환한다.

코드 06에서 09는 공간 객체의 확장된 그리드 선을 구한 후 변수의 속성에서 비트를 변환하고 하나씩 $bitmap_x$, $bitmap_y$ 에 문자열(String)로 저장한다. 최종 비트맵으로 변환된 좌표는 Python의 튜플(Tuple)을 사용하여 반환 한다(코드 10).

```

Algorithm : SpatialBitMap(Dataset, grid_numbers)
Dataset : geo-region of spatial objects
grid_numbers : The number of grids in x and y-axes
01: for x in Dataset:
02: (x_min, y_min), (x_max, y_max) ← Spatial objects' x and y-axes
03: grid_units_x = x_max / grid_numbers
04: grid_units_y = y_max / grid_numbers
05: gridline_x and gridline_y ← Object's x value / grid_units_x, if x
   less than a grid line, set gridline ← left line of the object. Or right
   line is set as gridline with floor or ceil function.
06: for bit in grid_units_x:
07: bitmap_x += bit as a string
08: for bit in grid_units_y:
09: bitmap_y += bit as a string
10: return (bitmap_x, bitmap_y)
    
```

Fig. 6. Bit-map Transformation Algorithm

본 연구에서는 4바이트 이상의 공간 데이터를 그리드로 변환된 공간에서 그리드 선을 비트로 변환함으로써 비트 단위로 저장한다. 비트로 변환된 좌표는 서버로부터 이동 단말기로 전송될 때 한 번에 많은 좌표를 전송하거나 하나의 좌표를 더 빠르게 전송하여 결과적으로는 서비스 제공 시간이 짧아진다.

IV. Simulation

제안기법은 공간 객체에 대한 네트워크 전송시간을 빠르게 하려고 객체의 좌표를 비트로 표현하여 이더넷 상에서 데이터의 전송속도를 실험하였다. 개발 및 시뮬레이션 환경은 <Table 1>과 같다.

Table 1. Simulation Environment

구분	내용
최대전송단위(MTU)	1500바이트
전송속도	가상 네트워크 환경에서 0.001 ms/bit
데이터 집합	가상 공간 데이터, 1만 개, 묶음은 단위를 K로 함.
데이터 분포	표준, 사향 분포
개발 언어 및 시스템	Python 3.2, RAM 128GB, XEON 2.8 GHz, Fedora 30

컴퓨터 네트워크를 대표하는 이더넷은 최대전송단위인 MTU가 1500바이트이다. MTU는 인터넷 프로토콜 5계층의 헤더를 포함하고 있으나 본 실험에서는 헤더의 크기는 고려하지 않고 순수하게 공간 데이터의 좌표만을 MTU 전체로 가정하여 실험하였다. 이는 실제 MTU에서 헤더를 추가하더라도 MTU에 포함되는 공간 객체의 수에 차이가 있을 뿐 개념적인 측면은 같으므로 실험 결과에는 영향을 미치지 않는다.

데이터 집합은 공간 객체 생성기를 이용한 1만 개의 실험용 객체를 생성하였으며 지리적 공간의 분포를 고려하여 표준 분포와 사향 분포에 대한 실험을 진행하였다. 네트워크는 가상 네트워크 환경을 구성하여 Python으로 구현하였으며 전송속도는 0.001 ms/bit 설정하였다.

가상의 실험 데이터는 Python으로 데이터 생성기를 작성하여 (Fig. 7)과 같이 표준 분포와 사향 분포의 공간 객체를 생성하였다. 두 분포를 비교하는 것은 공간 분할이 $n \times n$ 의 그리드로 되어 있는 상태에서 그리드 셀을 읽고 좌표를 비트맵으로 변환할 때 속도를 비교하기 위한 것이다. 즉, 변환 전후의 두 분포에서 변환된 속도 비교를 수행한다.

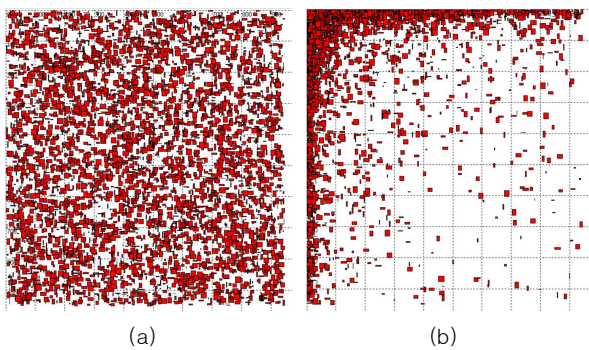


Fig. 7. Simulation Data Set, (a) Uniform, (b) Skewed

공간 객체는 MBR로 표현되므로 총 4개의 좌표를 사용하며 정수 4바이트로 가정하였을 때 16바이트를 차지한다. 이는 1500바이트의 표준 MTU에 93개의 위치 정보를 전송하게 된다. 또한, 각 좌표를 4bit에 저장할 때 한 객체는 2바이트로 표현되므로 750개의 공간 객체가 MTU에 저장되어 8배의 효율을 보인다. 다음의 (Fig. 8)은 MTU의 크기 변화에 따른 공간 객체의 수를 나타낸다.

(Fig. 9)는 표준 분포와 사향 분포에서 비트맵 변환 속도를 측정된 것이다. 표준 분포는 $n \times n$ 의 모든 그리드를 읽어서 공간 객체의 좌표를 비트맵으로 변환한다. 사향 분포도 모든 그리드를 읽는 것은 같지만 분포도에서 보는 것과 같이 공간 객체가 없는 셀도 존재하여 변환 속도는 표준 분포보다 빠른 것으로 실험되었다.

전체 공간 객체인 10K를 대상으로 할 때 사향 분포의 비트맵 변환 속도는 표준 분포보다 약 2.5배가 빠른 것으로 나타났다. 본 실험에서는 사향 분포의 경우 왼쪽 위에 객체가 모여있었지만, 전체적인 변환 속도에서 표준 분포보다 빠르게 처리되었다.

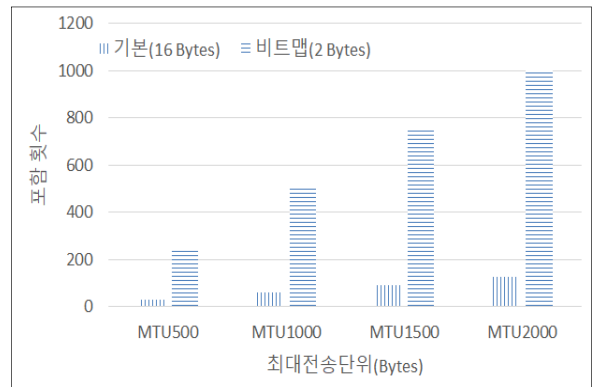


Fig. 8. Bit-map Transformation Algorithm

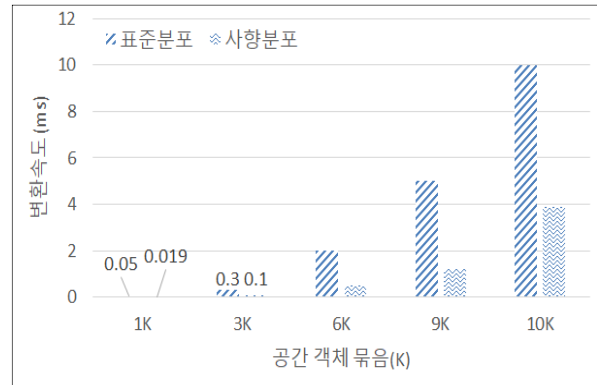


Fig. 9. Bit-map Transformation Algorithm

비트맵으로 변환이 완료된 후, 가상 네트워크에서 비트당 전송속도를 비교한 결과 (Fig. 10)과 같이 공간 객체의 묶음이 커질수록 차이가 벌어졌다. 이는 비트맵의 경우, MTU를 하나의 패킷으로 설정하여서 한 패킷에 들어가는 좌표의 수가 기본 좌표와 비교하면 약 8배의 차이가 있으므로 도착 지점으로 전송되는 전체 시간이 빠른 것으로 나타났다.

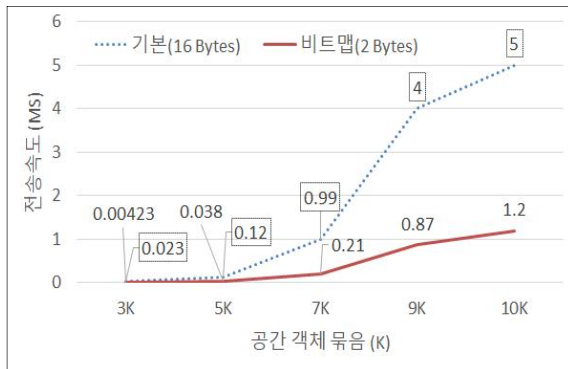


Fig. 10. Bit-map Transformation Algorithm

V. Conclusions

본 논문에서는 모바일 환경에서 공간 데이터의 생성과 사용이 증가하면서 네트워크를 통한 신속한 전송을 위한 비트맵 기반의 공간 데이터 전송기법을 제안하였다.

기존의 연구들이 공간 데이터를 이용한 서비스 제공에 초점을 맞춰서 공간 데이터에 대한 클러스터링 알고리즘 등을 사용하지만 최종 결과에 대한 전송속도는 고려하지 않았다.

본 연구에서는 공간 데이터가 인터넷 패킷을 통해 전송될 때 더 많은 비트가 포함되도록 공간 데이터를 비트맵 패턴으로 구성하였다. 실험에서는 16바이트로 구성되는 기본 값과 비트맵으로 변환된 공간 데이터를 산술 비교하였으며 가상의 데이터를 생성하여 변환 시간과 네트워크 전송시간을 비교하였다. 가상 데이터는 공간에서 읽어내는 속도의 차이를 비교하기 위해 표준 분포와 사향 분포로 생성하였으며, 실험 결과 비트맵 변환 시 2.5배, 네트워크 전송에서는 약 8배의 속도를 나타냈다.

논문에서 제안한 기법은 기존의 공간 데이터를 이용한 서비스 구성에 관한 연구와 더불어 사용하면 서비스 제공 속도 향상에 도움이 되며 향후, 다양한 공간 데이터를 사용하는 서비스에도 적용할 수 있어서 연구결과는 매우 고무적이며, 향후 연구방향으로는 16바이트 구성 기본 값과 비트맵 변환 공간데이터를 산술 비교를 넘어 실제 데이터의 전송속도로 측정하는 것이다.

REFERENCES

- [1] J. Schiller, A. Voisard, "Location-Based Services," Elsevier, 2004.
- [2] H. t. Hoi, V. H. Ca, L. V. Truong, "Location-Based Services," International Conference on Management, Economics, Business and Social Sciences, ICMEBSS, pp. 81-90, 2018.

- [3] M. Kibanov, M. Becker, J. Mueller, M. Atzmueller, A. Hotho, G. Stumme, "Adaptive kNN using expected accuracy for classification of geo-spatial data," Proceedings of the 33rd Annual ACM Symposium on Applied Computing, pp. 857-865, April 2018.
- [4] M. Najibi, M. Rastegari, L. S. Davis, "G-CNN: an iterative Grid Based Object Detector," The IEEE Conference on Computer Vision and Pattern Recognition, pp. 2369-2377, June 2016.
- [5] Q. Zhao, Y. Shi, Q. Liu, P. Franti, "A grid-growing clustering algorithm for geo-spatial data," Pattern Recognition Letters, Vol. 53, pp. 77-84, 2014.
- [6] K. Koperski, J. Han, N. Stefanovic, "An Efficient TwoStep Method for Classification of Spatial Data," SDH '98), Jan. 1999.
- [7] Y. Fang, R. Cheng, X. Li, S. Luo, J. Hu, "Effective community search over large spatial graphs," Proceedings of the VLDB Endowment, Vol. 10(6), pp. 709-720, February 2017.
- [8] T. Sajana, C. M. Sheela Rani, K. V. Narayana, "A Survey on Clustering Techniques for Big Data Mining," Indian Journal of Science and Technology, Vol. 9(3), pp. 1-12, January 2016.
- [9] A. Eldawy, L. Alarabi, M. F. Mokbel, "Spatial partitioning techniques in SpatialHadoop," International Conference on Very Large Data Bases, Vol. 8(12), August 2015.
- [10] H. J. Jang, B. Kim, J. Kim, S. Y. Jung, "An efficient grid-based k-prototypes algorithm for sustainable decision-making on spatial objects," Sustainability, Vol. 10(8), pp. 1-2, 2018.
- [11] A. Guttman, "R-trees: a dynamic index structure for spatial searching," in Proceedings of ACM SIGMOD International Conference on Management of Data, Vol. 14, pp. 47-57, 1984.

Authors



Gi Oug OH received the B.S. degree from Kyungwon University in Computer Engineering 1991. He received the M.S. and Ph.D. degrees in Computer Science and Engineering from Soongsil University, Korea, in 1993 and 2007, respectively.

He is currently a Assistant Professor in the Department of Computer Science, Gachon University. He is interested in spatial data change, sensor network, RFID and IOT authentication.